CVE-2018-1000224

Godot serialization security issues

Luiz Felipe Moumdjian Girotto
Victor Araujo

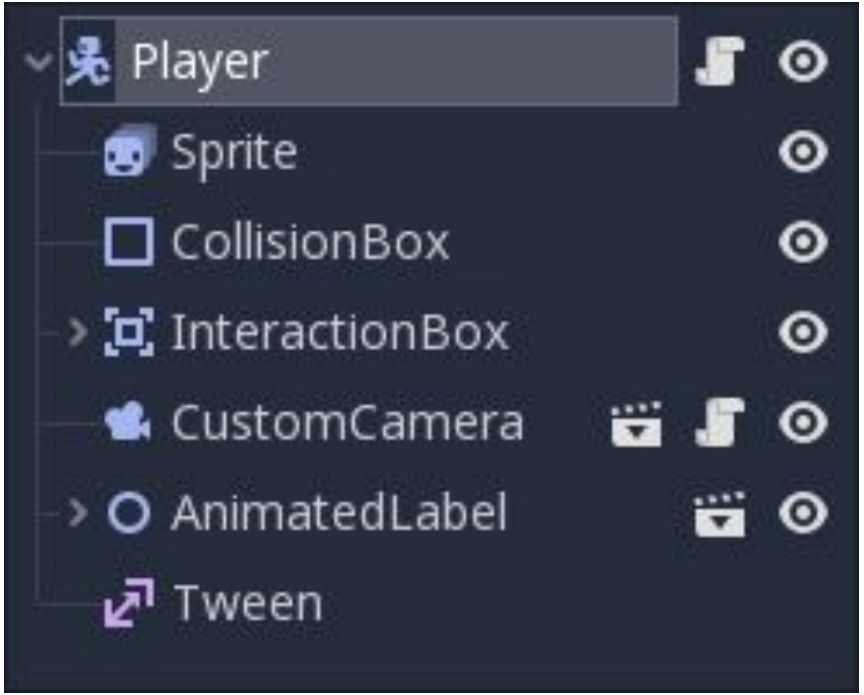
Introdução

Godot

- Uma *game engine*
- Criada em 2007, primeiro *release* e *open source* em 2014
- Versão 2.0 em 2016, e 3.0 em 2018, ambas mantidas até hoje

Godot

- Utilizada para criar jogos 2D e 3D
- Jogos construídos a partir de **nós** e **cenas**



Godot e *GDScript*

- Suporta oficialmente C, C++, *VisualScript*, e *GDScript*.

- *GDScript*: alto nível, tipagem dinâmica, alta integração com os elementos e estruturas da *engine*.

GDScript

- Possui *Variant* como tipo de dados atômico/nativo
- Todos outros tipos de dados *built-in* na linguagem dependem da existência do tipo *Variant*

O Problema

Godot e *Networking*

- Godot disponibiliza APIs para a criação de jogos multijogador *online*
- Disponibiliza APIs de baixo nível (TCP e UDP), mas também disponibiliza de alto nível

Networking

- A transmissão de dados por via destes protocolos de baixo nível requer a serialização (ou *marshalling*) dos dados a serem enviados

- Problemas no código de serialização até as versões 2.1.5 e 3.0.6 da Godot

Código de Serialização

- Problemas de *padding* causando o vazamento de memória interna pela rede

- Problemas na checagem de tamanho de certos tipos de dados levam a *engine* a tentar alocar grandes quantidades de memória, e ser fechada pelo SO

Código de Serialização

https://github.com/godotengine/godot/commit/5262d1bbcc81a06db66ac45c3f75535f231268bc

Demonstração

- Será demonstrado somente um dos problemas mencionados: o *packet of death*

- Foi feito um pequeno “jogo” para demonstrar este problema, que será rodado na versão 3.0.0 da Godot

A Correção

Código de Serialização

https://github.com/godotengine/godot/commit/5262d1bbcc81a06db66ac45c3f75535f231268bc

Demonstração

- Retornamos, então, ao mesmo “jogo”, com o mesmo código-fonte, mas agora rodando na versão 3.0.6 da Godot

- Esta foi a versão na qual foi divulgada a existência e resolução dos problemas mencionados

Ressalvas

- Por mais que o *packet of death* não consiga fechar o jogo, ele gera um erro e passa a responsabilidade para a *engine*, tornando o problema de outra natureza (auto-contido)

- Note que a Godot não possui nenhuma forma de error handling

Obrigado!